



# Red Hat Data Grid 8.2

## Data Grid Guide to Cross-Site Replication

グローバル Data Grid クラスターのデータのバックアップ



## Red Hat Data Grid 8.2 Data Grid Guide to Cross-Site Replication

---

グローバル Data Grid クラスターのデータのバックアップ

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Data\_Grid\_Guide\_to\_Cross-Site\_Replication.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

Data Grid は、グローバルクラスターを形成して、地理的な場所全体にデータを複製できます。本ガイドでは、キャッシュのバックアップの場所を設定し、クロスサイト操作を実行する方法を説明します。また、グローバルクラスターの監視およびトラブルシューティングの方法についても説明します。

## 目次

RED HAT DATA GRID .....	4
DATA GRID のドキュメント .....	5
DATA GRID のダウンロード .....	6
多様性を受け入れるオープンソースの強化 .....	7
<b>第1章 DATA GRID クロスサイトレプリケーション .....</b>	<b>8</b>
1.1. クロスサイトレプリケーション .....	8
1.1.1. サイトマスター .....	8
1.2. キャッシュへのバックアップの追加 .....	9
1.3. バックアップストラテジー .....	9
同期ストラテジー .....	9
非同期ストラテジー .....	10
同期バックアップと非同期バックアップの比較 .....	10
1.4. バックアップを自動的にオフラインにする .....	10
1.5. 状態遷移 .....	11
1.5.1. 自動状態遷移 .....	12
1.6. サイト全体でのクライアント接続 .....	12
Active/Passive .....	12
Active/Active .....	13
バックアップストラテジー .....	13
1.6.1. 同時書き込みおよび競合エントリー .....	13
競合解決アルゴリズム .....	14
1.7. クロスサイトレプリケーションを使用した有効期限 .....	15
<b>第2章 クロスサイトレプリケーションのための DATA GRID の設定 .....</b>	<b>16</b>
2.1. クロスサイトレプリケーションのためのクラスタートランスポートの設定 .....	16
2.1.1. JGroups RELAY2 スタック .....	16
2.1.2. カスタム JGroups RELAY2 スタック .....	17
2.2. キャッシュへのバックアップの場所の追加 .....	18
2.3. 異なる名前でのキャッシュへのバックアップ .....	19
2.4. サイト間の状態遷移の設定 .....	19
2.5. 競合解決アルゴリズムのカスタマイズ .....	20
2.6. クロスサイトビューの確認 .....	21
2.7. クロスサイトレプリケーション用の HOT ROD クライアントの設定 .....	21
<b>第3章 クロスサイトレプリケーション操作の実行 .....</b>	<b>23</b>
3.1. CLI を使用したクロスサイト操作の実行 .....	23
3.1.1. バックアップ場所のオフラインおよびオンライン化 .....	23
3.1.2. サイト間の状態転送モードの設定 .....	23
3.1.3. バックアップ場所への状態のプッシュ .....	24
3.2. REST API を使用したクロスサイト操作の実行 .....	24
3.2.1. すべてのバックアップロケーションのステータス取得 .....	24
3.2.2. 特定のバックアップ場所のステータスの取得 .....	25
3.2.3. バックアップ先をオフラインにする .....	25
3.2.4. バックアップ場所をオンラインにする .....	26
3.2.5. バックアップ場所への状態のプッシュ .....	26
3.2.6. 状態転送のキャンセル .....	26
3.2.7. 状態転送ステータスの取得 .....	26
3.2.8. 状態転送ステータスのクリア .....	26
3.2.9. オフラインテイク条件の変更 .....	27

3.2.10. 受信サイトからの状態転送のキャンセル	27
3.2.11. バックアップ場所のステータスの取得	27
3.2.12. バックアップ先をオフラインにする	28
3.2.13. バックアップ場所をオンラインにする	28
3.2.14. 状態転送モードの取得	28
3.2.15. 状態転送モードの設定	28
3.2.16. 状態転送の開始	29
3.2.17. 状態転送のキャンセル	29
3.3. JMX を使用したクロスサイト操作の実行	29
3.3.1. JMX MBean を登録するための Data Grid の設定	29
3.3.2. クロスサイト操作の実行	29
<b>第4章 グローバル DATA GRID クラスターの監視およびトラブルシューティング</b>	<b>31</b>
4.1. DATA GRID 統計の有効化	31
4.2. DATA GRID メトリクスの設定	31
4.2.1. Data Grid メトリクスの収集	32
4.3. JMX MBEAN を登録するための DATA GRID の設定	33
4.3.1. クロスサイトレプリケーション用の JMX MBean	33
4.4. ログの収集およびクロスサイトレプリケーションのトラブルシューティング	34
4.4.1. クロスサイトログメッセージ	34



## RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

### スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

### グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

### エラスティックスケールリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

### データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。



## DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- [Data Grid 8.2 ドキュメント](#)
- [Data Grid 8.2 コンポーネントの詳細](#)
- [Data Grid 8.2 でサポートされる設定](#)
- [Data Grid 8 機能のサポート](#)
- [Data Grid で非推奨の機能](#)

## DATA GRID のダウンロード

Red Hat カスタマーポータルで [Data Grid Software Downloads](#) にアクセスします。



### 注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社の CTO である Chris Wright のメッセージ](#) を参照してください。

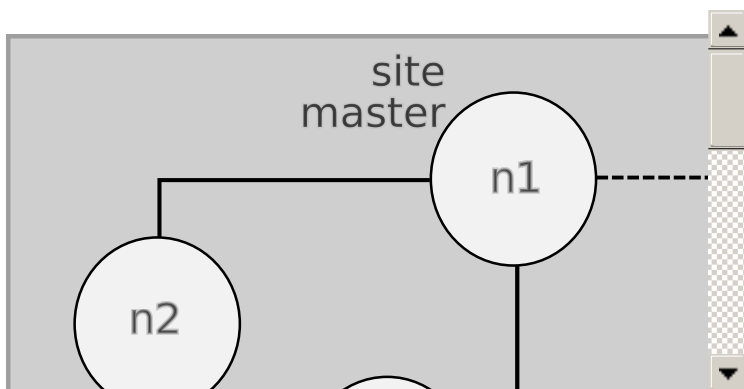
## 第1章 DATA GRID クロスサイトレプリケーション

クロスサイトレプリケーションを使用すると、ある Data Grid クラスタから別の Data Grid クラスタにデータのバックアップを作成できます。クラスタを設定する前に、Data Grid クロスサイトレプリケーションの仕組みを理解するための概念を説明します。

### 1.1. クロスサイトレプリケーション

異なる場所で実行されている Data Grid クラスタは、相互に検出および通信できます。

サイトは通常、地理的にさまざまな場所にあるデータセンターです。以下の図のように、クロスサイトレプリケーションはサイト内の Data Grid クラスタをブリッジし、グローバルクラスタを形成します。



**LON** は、イギリスのロンドンにあるデータセンターです。  
**NYC** は、米国のニューヨーク市にあるデータセンターです。



#### 注記

Data Grid は、2 つ以上のサイトにまたがってグローバルクラスタを形成できます。

たとえば、**LON** および **NYC** のバックアップ場所として、サンフランシスコで実行している 3 番目の Data Grid クラスタ **SFO** を設定します。

#### 1.1.1. サイトマスター

サイトマスターは、Data Grid クラスタのノードであり、バックアップの場所から要求を送受信します。

ノードがサイトマスターでない場合は、バックアップ要求をローカルサイトマスターに転送する必要があります。バックアップの場所に要求を送信できるのは、サイトマスターだけです。

最適なパフォーマンスを得るには、すべてのノードをサイトマスターとして設定する必要があります。これにより、クラスタ内の各ノードがバックアップ要求をサイトマスターに転送しなくても、リモートサイトに直接バックアップできるため、バックアップ要求の速度が向上します。



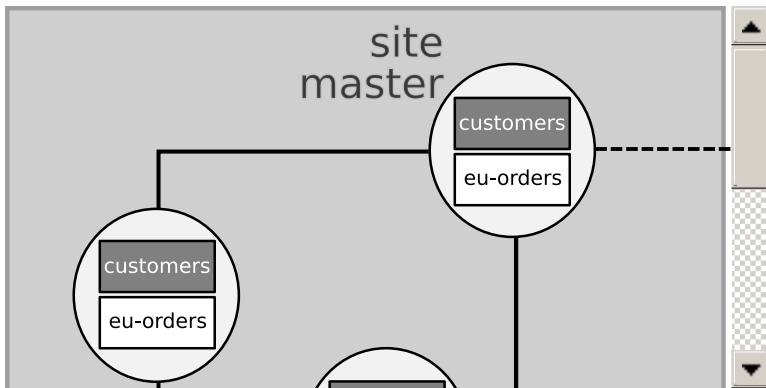
#### 注記

このドキュメントの図では、JGroups RELAY2 プロトコルのデフォルトが 1 つのサイトマスターを持つ Data Grid クラスタであることから、これを示しています。同様に、クラスタ内の各サイトマスターはリモートクラスタ内の各サイトマスターと通信するため、単一のサイトマスターの方が説明が簡単です。

## 1.2. キャッシュへのバックアップの追加

キャッシュ定義でバックアップの場所としてリモートサイトに名前を付けます。

たとえば、以下の図では、3つのキャッシュ "customers"、"eu-orders"、および "us-orders" を示しています。



- LON では、"customers" はバックアップの場所として NYC を指定します。
- NYC では、"customers" はバックアップの場所として LON を指定します。
- "eu-orders" および "us-orders" にはバックアップはなく、それぞれのクラスターのローカルになります。

## 1.3. バックアップストラテジー

Data Grid は、キャッシュへの書き込みが発生すると同時に、クラスター間でデータを複製します。たとえば、クライアントが LON に "k1" と書き込む場合、Data Grid は同時に "k1" を NYC バックアップします。

別のクラスターにデータのバックアップを作成するには、Data Grid は同期または非同期ストラテジーを使用します。

### 同期ストラテジー

Data Grid がデータをバックアップの場所に複製するとき、ローカルクラスターのキャッシュとリモートクラスターのキャッシュに同時に書き込みます。同期ストラテジーでは、Data Grid は両方の書き込み操作が完了するのを待ってから戻ります。

バックアップ操作が失敗した場合に、Data Grid がローカルクラスターのキャッシュへの書き込みを処理する方法を制御できます。Data Grid は以下を行うことができます。

- 失敗したバックアップを無視し、ローカルクラスターへの書き込みを警告せずに続行します。
- 警告メッセージをログに記録するか、例外を出力し、ローカルクラスターへの書き込みを続行します。
- 失敗したバックアップ操作をカスタムロジックで処理します。

同期バックアップは、楽観的なトランザクションに参加するキャッシュを持つ 2 フェーズコミットもサポートします。バックアップの最初のフェーズはロックを取得します。2 番目のフェーズは変更をコミットします。



## 重要

クロスサイトレプリケーションのある 2 フェーズコミットは、ネットワーク全体で 2 つのラウンドトリップが必要なため、パフォーマンスに大きく影響します。

### 非同期ストラテジー

Data Grid がデータをバックアップ場所に複製する場合、操作が完了するのを待たずにローカルキャッシュに書き込みます。

非同期バックアップ操作およびローカルキャッシュへの書き込みは、互いに独立しています。バックアップ操作が失敗した場合は、ローカルキャッシュへの書き込み操作は続行され、例外は発生しません。これが発生すると、Data Grid は、リモートクラスターがクロスサイトビューから切断されるまで、書き込み操作も再試行します。

### 同期バックアップと非同期バックアップの比較

同期バックアップは、サイト全体でのデータの一貫性を最も強力に保証します。**strategy=sync** の場合、**cache.put()** 呼び出しが返されると、ローカルキャッシュとバックアップの場所で値が最新の状態であることがわかります。

この一貫性のために犠牲となるのがパフォーマンスです。同期バックアップは、非同期バックアップと比較すると、レイテンシーが大幅に高くなります。

一方、非同期バックアップは、クライアントリクエストにレイテンシーを追加しないので、パフォーマンスに影響を及ぼすことはありません。ただし、**strategy=async** の場合、**cache.put()** 呼び出しが返されると、バックアップの場所の値がローカルキャッシュの値と同じであるかどうかを確認できません。

## 1.4. バックアップを自動的にオフラインにする

リモートサイトが利用できなくなった場合に、バックアップの場所が自動的にオフラインになるように設定できます。これにより、Data Grid ノードが、オフラインのバックアップ場所にデータを継続的に複製しようとして、エラーメッセージが表示されたり、リソースが消費されたりすることを防ぎます。

### バックアップ操作のタイムアウト

バックアップ設定には、データを複製する操作のタイムアウト値が含まれます。タイムアウトが発生する前に操作が完了しない場合、Data Grid は失敗としてそれらを記録します。

```
<!-- Operations to replicate data to NYC are recorded as failures if they do not complete after 10 seconds. -->
<backup site="NYC" strategy="ASYNC" timeout="10000">
  <!-- Configuration for backup locations goes here. -->
</backup>
```

### 失敗の数

バックアップの場所がオフラインになる前に発生する可能性のある **連続する** 失敗の数を指定できます。

たとえば、NYC の以下の設定は、オフラインになる前に失敗する操作の数を 5 に設定しています。

```
<!-- If a cluster attempts to replicate data to NYC and five consecutive operations fail, NYC automatically goes offline. -->
<backup site="NYC" strategy="ASYNC" timeout="10000">
  <take-offline after-failures="5"/>
</backup>
```

## 待機時間

バックアップ操作の失敗時に、サイトをオフラインにするまで待機する時間を指定することもできます。待機時間がなくなる前にバックアップ要求が成功した場合、Data Grid はサイトをオフラインにしません。

```
<!-- If a cluster attempts to replicate data to NYC and there are more than five consecutive failures
and 15 seconds elapse after the first failed operation, NYC automatically goes offline. -->
<backup site="NYC" strategy="ASYNC" timeout="10000">
  <take-offline after-failures="5"
    min-wait="15000"/>
</backup>
```

## ヒント

サイトがオフラインになるまでの待機時間を最小限にしたい場合は、**after-failures** 属性に負またはゼロの値を設定します。

```
<take-offline after-failures="-1" min-wait="10000"/>
```

## 1.5. 状態遷移

状態遷移は、サイト間でデータを同期する管理操作です。

たとえば、LON がオフラインになると、NYC がクライアント要求の処理を開始します。LON をオンラインに戻すと、LON の Data Grid クラスターには NYC のクラスターと同じデータはありません。

LON と NYC の間でデータの一貫性を保つには、NYC から LON に状態をプッシュできます。

- 状態遷移は双方向です。たとえば、NYC から LON へ、または LON から NYC へ状態のプッシュを実行できます。
- 状態をオフラインサイトにプッシュすると、オンラインに戻ります。
- 状態転送は、発信元サイトと受信サイトの両方のサイトに存在するデータのみを上書きしません。Data Grid はデータを削除しません。  
たとえば、"k2" は LON および NYC に存在します。"k2" は、LON がオフライン時に NYC から削除されます。LON をオンラインに戻すと、"k2" は引き続きその場所に存在します。NYC から LON に状態をプッシュすると、転送は LON の "k2" には影響しません。

## ヒント

状態遷移後にキャッシュの内容が同じになるようにするには、状態をプッシュする前に受信サイトのキャッシュからすべてのデータを削除します。

CLI から **clear()** メソッドまたは **clearcache** コマンドを使用します。

- 状態遷移は、プッシュの開始後に発生するデータへの更新を上書きしません。  
たとえば、"k1,v1" は LON および NYC に存在します。LON がオフラインになったため、NYC から LON に状態転送をプッシュします。これにより、LON が再びオンラインになります。状態遷移が完了する前に、クライアントは LON に "k1,v2" を配置します。

この場合、プッシュを開始した後に変更が発生したため、NYC からの状態遷移は "k1,v2" を上書きしません。

### 1.5.1. 自動状態遷移

デフォルトでは、CLI を使用して、または JMX や REST 経由で、クロスサイトの状態遷移操作を手動で実行する必要があります。

ただし、非同期バックアップストラテジーを使用すると、Data Grid はクロスサイト状態遷移操作を自動的に実行できます。バックアップの場所がオンラインになり、ネットワーク接続が安定していることを検知すると、Data Grid はバックアップの場所の間で双方向状態遷移を開始します。たとえば、Data Grid は、状態を **LON** から **NYC** に、**NYC** から **LON** に同時に遷移します。



#### 注記

一時的なネットワークの切断による状態転送操作のトリガーを回避するために、バックアップの場所がオフラインになるために満たす必要のある条件が2つあります。バックアップの場所のステータスはオフラインでなければならず、JGroups RELAY2 のクロスサイトビューに含めることはできません。

#### 参照資料

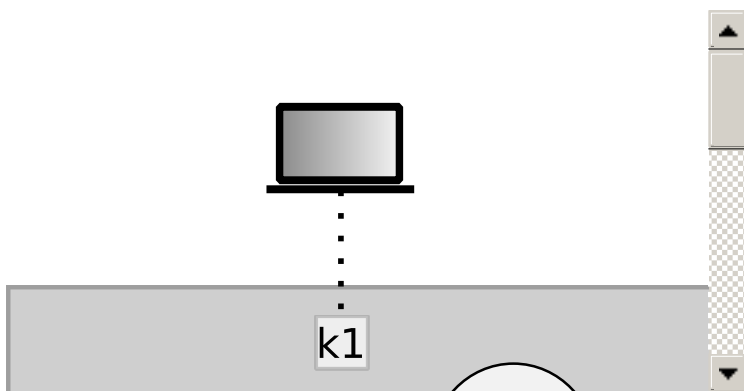
- [org.infinispan.Cache.clear\(\)](#)
- [Clearing Caches with the CLI](#)
- [Clearing Caches with the REST API](#)
- クロスサイト状態遷移の設定

## 1.6. サイト全体でのクライアント接続

クライアントは、Active/Passive または Active/Active 設定のいずれかで Data Grid クラスタに書き込みできます。

#### Active/Passive

以下の図は、Data Grid が1つのサイトのみからのクライアント要求を処理する Active/Passive を示しています。



上記のイメージでは、以下ようになります。

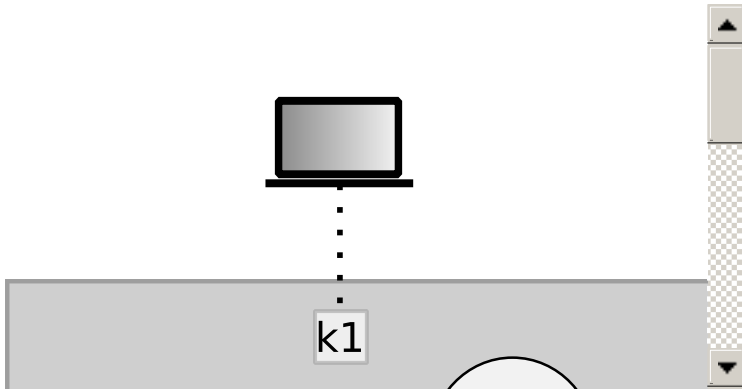
1. クライアントは **LON** で Data Grid クラスタに接続します。
2. クライアントは "k1" をキャッシュに書き込みます。
3. **LON** のサイトマスター n1 は、k1 をレプリケートする要求を **NYC** のサイトマスター nA に送信します。



Active/Passive では、NYC はデータの冗長性を提供します。LON の Data Grid クラスタが何らかの理由でオフラインになると、クライアントは NYC への要求の送信を開始できます。LON をオンラインに戻すと、NYC とデータを同期し、クライアントを LON に戻すことができます。

### Active/Active

以下の図は、Data Grid が 2 つのサイトでクライアント要求を処理する Active/Active を示しています。

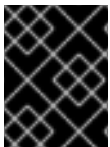


上記のイメージでは、以下のようになります。

1. クライアント A は、LON で Data Grid クラスタに接続します。
2. クライアント A は "k1" をキャッシュに書き込みます。
3. クライアント B は、NYC の Data Grid クラスタに接続します。
4. クライアント B は "k2" をキャッシュに書き込みます。
5. LON および NYC のサイトマスターは、"k1" が NYC に複製され、"k2" が LON に複製されるように要求を送信します。

Active/Active では、NYC と LON の両方は、クライアント要求の処理中にデータをリモートキャッシュに複製します。NYC または LON のいずれかがオフラインになると、クライアントはオンラインサイトへの要求の送信を開始できます。その後、オフラインサイトをオンラインに戻し、状態をプッシュしてデータを同期し、必要に応じてクライアントを切り替えることができます。

### バックアップストラテジー



#### 重要

Active/Active 設定では、常に非同期バックアップストラテジー (**strategy=async**) を使用する必要があります。

複数のクライアントが同じエントリーに同時に書き込もうとし、バックアップストラテジーが同期 (**strategy=sync**) の場合、デッドロックが発生します。ただし、両方のサイトが異なるデータセットにアクセスする場合、Active/Passive 設定で同期バックアップストラテジーを使用できます。この場合、同時書き込みによるデッドロックのリスクはありません。

#### 1.6.1. 同時書き込みおよび競合エントリー

クライアントが同時に同じエントリーに書き込むが、サイトが異なる場合は、Active/Active サイト設定でエントリーの競合が発生する可能性があります。

たとえば、クライアント B が NYC の k1 に書き込むのと同時に、クライアント A は LON の "k1" に書き込みます。この場合、"k1" の値は LON と NYC とで異なります。レプリケーションの発生後、"k1" のどの値がどのサイトに存在するかは保証されません。

データの整合性を確保するために、Data Grid はベクトルロックアルゴリズムを使用して、次の図のように、バックアップ操作中に競合するエントリーを検出します。

	LON	NYC
k1=(n/a)	0,0	0,0
k1=2	1,0	1,0
k1=3	1,1	1,1
k1=5	2,1	1,2
	--> 2,1 (conflict)	
(conflict)	1,2	<--

ベクトルロックは、エントリーへの書き込みごとにインクリメントするタイムスタンプのメタデータです。上記の例では、**0,0** は、"k1" 上のベクトルロックの初期値を表します。

クライアントは LON に "k1=2" を配置し、ベクトルロックは **1,0** で、Data Grid はこれを NYC に複製します。その後、クライアントは NYC に "k1=3" を配置し、ベクトルロックは **1,1** に更新されます。Data Grid はこれを LON に複製します。

ただし、クライアントが NYC に "k1=8" を配置するのと同時に、クライアントが LON に "k1=5" を配置すると、Data Grid は競合するエントリーを検出します。これは、"k1" のベクター値が LON と NYC の間で厳密に大きかったり小さかったりしないためです。

競合するエントリーを見つけると、Data Grid は Java `compareTo(String anotherString)` メソッドを使用してサイト名を比較します。どのキーが優先されるかを決定するために、Data Grid は辞書式順序で他のキーよりも小さいサイト名を選択します。**AAA** という名前のサイトからのキーは、**AAB** という名前のサイトからのキーよりも優先されます。

同じ例に従って、"k1" の競合を解決するために、Data Grid は LON からの "k1" の値を使用します。これにより、Data Grid が競合を解決し、値を複製した後に、LON と NYC の両方で "k1=5" となります。

## ヒント

競合するエントリーを解決するための優先順位を表す簡単な方法として、サイト名の前に番号を付けます。たとえば、**1LON** や **2NYC** などです。



### 注記

Data Grid は、非同期バックアップストラテジ (`strategy=async`) のみを使用して競合解決を実行します。ただし、同時書き込みはデッドロックになるため、Active/Active 設定で同期バックアップストラテジを使用しないでください。

### 競合解決アルゴリズム

Data Grid は、カスタム競合解決ストラテジを実装できる `XSiteEntryMergePolicy` SPI に加えて、競合を解決するためのさまざまなアルゴリズムを提供します。

辞書式比較を使用するデフォルトの競合解決戦略とは別に、Data Grid の競合解決アルゴリズムを使用して次のことができます。

- 競合するエントリーを常に削除します。
- 書き込み/削除の競合が発生したときに書き込み操作を保持します。
- 書き込み/削除の競合が発生したときにエントリーを削除します。

**org.infinispan.xsite.spi.XSiteMergePolicy** 列挙で、利用可能なすべてのアルゴリズムとその説明を検索します。

#### 関連情報

- [java.lang.String#compareTo\(\)](#)
- [org.infinispan.xsite.spi.XSiteMergePolicy](#)
- [org.infinispan.xsite.spi.XSiteEntryMergePolicy](#)
- [競合解決戦略のカスタマイズ](#)

## 1.7. クロスサイトレプリケーションを使用した有効期限

有効期限は、時間に基づいてキャッシュエントリーを削除します。Data Grid では、エントリーの有効期限を設定する方法は2つあります。

#### 有効期間

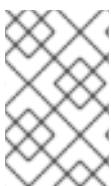
エントリーが存在することができる最大時間を設定します。

クロスサイトレプリケーションで **lifespan** を設定する場合、Data Grid クラスターはリモートサイトとは別にエントリーを期限切れにします。

#### max-idle

特定の期間における読み取り操作または書き込み操作に基づいて、エントリーが存在する期間を指定します。

クロスサイトレプリケーションで **max-idle** を設定すると、Data Grid クラスターは touch コマンドを送信して、リモートサイトでアイドルタイムアウト値を調整します。



#### 注記

クロスサイトデプロイメントで最大アイドル有効期限を使用すると、**max-idle** 値の同期を維持するための追加の処理により、一部の操作が完了するまで時間がかかることがあるため、パフォーマンスに影響を及ぼす可能性があります。

## 第2章 クロスサイトレプリケーションのための DATA GRID の設定

サイト間でデータをレプリケートするように Data Grid を設定するには、最初にクラスタートランスポートをセットアップして、Data Grid クラスターが相互に検出し、サイトマスターが通信できるようにします。次に、バックアップの場所を Data Grid 設定のキャッシュ定義に追加します。

### 2.1. クロスサイトレプリケーションのためのクラスタートランスポートの設定

Data Grid クラスターがバックアップの場所と通信できるように、JGroups RELAY2 をトランスポート層に追加します。

#### 手順

1. `infinispan.xml` を開いて編集します。
2. RELAY2 プロトコルを JGroups スタックに追加します。
3. スタックを使用するように Data Grid クラスタートランスポートを設定します。

```
<infinispan>
  <jgroups>
    <!-- Extends the default UDP stack. -->
    <stack name="xsite" extends="udp">
      <!-- Adds RELAY2 for cross-site replication. -->
      <!-- Names the local site as LON. -->
      <!-- Specifies 1000 nodes as the maximum number of site masters. -->
      <relay.RELAY2 site="LON" xmlns="urn:org:jgroups" max_site_masters="1000"/>
      <!-- Uses the default TCP stack for inter-cluster communication. -->
      <!-- Names all sites that act as backup locations. -->
      <remote-sites default-stack="tcp">
        <remote-site name="LON"/>
        <remote-site name="NYC"/>
      </remote-sites>
    </stack>
  </jgroups>
  <cache-container name="default" statistics="true">
    <!-- Use the "xsite" stack for cluster transport. -->
    <transport cluster="{cluster.name}" stack="xsite"/>
  </cache-container>
</infinispan>
```

4. `infinispan.xml` を保存して閉じます。

#### 参照資料

- [JGroups RELAY2 スタック](#)
- [Data Grid 設定スキーマ](#)

#### 2.1.1. JGroups RELAY2 スタック

Data Grid クラスターは、クラスター間の検出および通信に JGroups RELAY2 を使用します。

```

<jgroups>
  <!-- Uses the default JGroups UDP stack for intra-cluster traffic. -->
  <stack name="xsite" extends="udp">
    <!-- Adds RELAY2 to the stack for inter-cluster transport. -->
    <!-- Names the local site. Data in caches from the local site is replicated to backup locations. -->
    <!-- Configures a maximum of 1000 site masters for the local cluster. -->
    <relay.RELAY2 xmlns="urn:org:jgroups"
      site="LON"
      max_site_masters="1000"/>
    <!-- Specifies all site names and uses the default JGroups TCP stack for inter-cluster transport. -->
  >
  <remote-sites default-stack="tcp">
    <!-- Names all sites that participate in cross-site replication. -->
    <remote-site name="LON"/>
    <remote-site name="NYC"/>
  </remote-sites>
</stack>
</jgroups>

```

## ヒント

バックアップ要求で最適なパフォーマンスを得るには、**max\_site\_masters** に Data Grid クラスタ内のノード数以上の値を設定します。

### 2.1.2. カスタム JGroups RELAY2 スタック

以下の設定は、デフォルトの TCP スタックを拡張するカスタムの RELAY2 スタックを追加します。

```

<jgroups>
  <!-- Uses TCPping instead of MPING for discovery. -->
  <stack name="relay-global" extends="tcp">
    <MPING stack.combine="REMOVE"/>
    <TCPping initial_hosts="192.0.2.0[7800]"
      stack.combine="INSERT_AFTER"
      stack.position="TCP"/>
  </stack>
  <!-- Extends the default UDP stack with RELAY2. -->
  <!-- Specifies RELAY2 properties. -->
  <stack name="xsite" extends="udp">
    <relay.RELAY2 site="LON" xmlns="urn:org:jgroups"
      max_site_masters="10"
      can_become_site_master="true"/>
    <remote-sites default-stack="relay-global">
      <remote-site name="LON"/>
      <remote-site name="NYC"/>
    </remote-sites>
  </stack>
</jgroups>

```

また、外部で定義された JGroups スタックファイルを参照することもできます。以下に例を示します。

```

<infinispan>
  <jgroups>

```

```

    <stack-file name="relay-global" path="jgroups-relay.xml"/>
  </jgroups>
  <cache-container default-cache="replicatedCache">
    <transport stack="relay-global" />
    <replicated-cache name="replicatedCache"/>
  </cache-container>
</infinispan>

```

**jgroups-relay.xml** は、以下のような JGroups スタックファイルを参照します。

```

<config xmlns="urn:jgroups"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:jgroups http://www.jgroups.org/schema/jgroups-4.1.xsd">

  <!-- Use TCP for inter-cluster transport. -->
  <TCP bind_addr="127.0.0.1"
    bind_port="7200"
    port_range="30"

    thread_pool.min_threads="0"
    thread_pool.max_threads="8"
    thread_pool.keep_alive_time="5000"
  />

  <!-- Use TCPPING for inter-cluster discovery. -->
  <TCPPING timeout="3000"
    initial_hosts="127.0.0.1[7200]"
    port_range="3"
    ergonomics="false"/>

  <!-- Provide other JGroups stack configuration as required. -->
</config>

```

## 参照資料

- [Setting Up Data Grid Clusters](#)
- [JGroups RELAY2](#)
- [複数のサイト間のリレー \(RELAY2\)](#)

## 2.2. キャッシュへのバックアップの場所の追加

Data Grid がデータをそれらの場所にバックアップできるように、リモートサイトの名前を指定します。

### 手順

1. **backups** 要素をキャッシュ定義に追加します。
2. 各リモートサイトの名前を **backup** 要素で指定します。  
たとえば、LON 設定で、NYC をリモートサイトとして指定します。

3. 各サイトが他のすべてのサイトのバックアップとなるように、前述の手順を繰り返します。たとえば、NYC を LON のバックアップとして追加せずに、LON を NYC のバックアップとして追加することはできません。



### 注記

キャッシュ設定はサイト間で異なり、異なるバックアップストラテジーを使用する場合があります。Data Grid は、キャッシュ名に基づいてデータを複製します。

### LONでの "customers" の設定例

```
<replicated-cache name="customers">
  <backups>
    <backup site="NYC" strategy="ASYNC" />
  </backups>
</replicated-cache>
```

### NYCでの "customers" の設定例

```
<distributed-cache name="customers">
  <backups>
    <backup site="LON" strategy="SYNC" />
  </backups>
</distributed-cache>
```

### 参照資料

- [Data Grid 設定スキーマ](#)

## 2.3. 異なる名前でのキャッシュへのバックアップ

デフォルトでは、Data Grid は同じ名前のキャッシュ間でデータを複製します。

### 手順

- **backup-for** を使用して、リモートサイトからのデータをローカルサイトの別の名前でのキャッシュに複製します。

たとえば、以下の設定は、LON の "customers" キャッシュを NYC の "eu-customers" キャッシュにバックアップします。

```
<distributed-cache name="eu-customers">
  <backups>
    <backup site="LON" strategy="SYNC" />
  </backups>
  <backup-for remote-cache="customers" remote-site="LON" />
</distributed-cache>
```

## 2.4. サイト間の状態遷移の設定

クロスサイト状態遷移設定を変更し、パフォーマンスを最適化して、操作を手動で行うか自動で行うかを指定します。

## 手順

- `<state-transfer>` を使用して、状態遷移操作を設定します。

たとえば、以下の設定は、カスタム設定を使用してクロスサイト状態遷移を自動的に実行します。

```
<distributed-cache name="eu-customers">
  <backups>
    <backup site="LON" strategy="ASYNC">
      <state-transfer chunk-size="64"
        timeout="30000"
        max-retries="30"
        wait-time="2000"
        mode="AUTO"/>
    </backup>
  </backups>
</distributed-cache>
```

## 2.5. 競合解決アルゴリズムのカスタマイズ

別のアルゴリズムを使用してバックアップの場所間の競合するエントリーを解決するように Data Grid を設定します。

### 手順

1. Data Grid 設定を開いて編集します。
2. Data Grid アルゴリズムまたはカスタム実装のいずれかを、**backups** 要素の **merge-policy** 属性で指定します。

### Data Grid アルゴリズムの使用

**org.infinispan.xsite.spi.XSiteMergePolicy** 列挙で、すべての Data Grid アルゴリズムと説明を検索します。

以下の設定例では、両方のサイトから競合するエントリーを削除する **ALWAYS\_REMOVE** アルゴリズムを使用します。

```
<distributed-cache name="eu-customers">
  <backups merge-policy="ALWAYS_REMOVE">
    <backup site="LON" strategy="ASYNC"/>
  </backups>
</distributed-cache>
```

### カスタム実装の使用

1. カスタムの **XSiteEntryMergePolicy** 実装を作成します。

```
public interface XSiteEntryMergePolicy<K, V> {
    CompletionStage<SiteEntry<V>> merge(K key, SiteEntry<V> localEntry, SiteEntry<V>
remoteEntry);
}
```

2. 以下の例のように、**merge-policy** 属性の値として完全修飾クラス名を指定します。



```
<distributed-cache name="eu-customers">
  <backups merge-policy="org.mycompany.MyCustomXSiteEntryMergePolicy">
    <backup site="LON" strategy="ASYNC"/>
  </backups>
</distributed-cache>
```

## 関連情報

- [org.infinispan.xsite.spi.XSiteEntryMergePolicy](#)
- [org.infinispan.xsite.spi.XSiteMergePolicy](#)
- [org.infinispan.xsite.spi.SiteEntry](#)
- [Data Grid 設定スキーマ](#)

## 2.6. クロスサイトビューの確認

クロスサイトレプリケーション用に Data Grid を設定した後、Data Grid クラスターがクロスサイトビューを正常に形成することを確認する必要があります。

### 手順

- ログメッセージ **ISPN000439: Received new x-site view** を確認してください。

たとえば、**LON** の Data Grid クラスターが、**NYC** の Data Grid クラスターを使用してクロスサイトビューを形成した場合、以下のメッセージが提供されます。

```
INFO [org.infinispan.XSITE] (jgroups-5,${server.hostname}) ISPN000439: Received new x-site view:
[NYC]
INFO [org.infinispan.XSITE] (jgroups-7,${server.hostname}) ISPN000439: Received new x-site view:
[NYC, LON]
```

## 2.7. クロスサイトレプリケーション用の HOT ROD クライアントの設定

異なるサイトで Data Grid クラスターを使用するように Hot Rod クライアントを設定します。

### hotrod-client.properties

```
# Servers at the active site
infinispan.client.hotrod.server_list = LON_host1:11222,LON_host2:11222,LON_host3:11222

# Servers at the backup site
infinispan.client.hotrod.cluster.NYC =
NYC_hostA:11222,NYC_hostB:11222,NYC_hostC:11222,NYC_hostD:11222
```

### ConfigurationBuilder

```
ConfigurationBuilder builder = new ConfigurationBuilder();
builder.addServers("LON_host1:11222;LON_host2:11222;LON_host3:11222")
    .addCluster("NYC")
```

```
.addClusterNodes("NYC_hostA:11222;NYC_hostB:11222;NYC_hostC:11222;NYC_hostD:11222")
```

## ヒント

以下の方法を使用して、Hot Rod クライアントをデフォルトのクラスターまたは別のサイトのクラスターに切り替えます。

- **RemoteCacheManager.switchToDefaultCluster()**
- **RemoteCacheManager.switchToCluster(\${site.name})**

## 参照資料

- [org.infinispan.client.hotrod.configuration package description](#)
- [org.infinispan.client.hotrod.configuration.ConfigurationBuilder](#)
- [org.infinispan.client.hotrod.RemoteCacheManager](#)

## 第3章 クロスサイトレプリケーション操作の実行

サイトをオンラインとオフラインにします。キャッシュ状態をリモートサイトに転送します。

### 3.1. CLI を使用したクロスサイト操作の実行

Data Grid コマンドラインインターフェイスを使用すると、Data Grid サーバーにリモートで接続したり、サイトを管理したり、状態遷移をバックアップの場所にプッシュしたりできます。

#### 前提条件

- Data Grid CLI を起動している。
- 実行中の Data Grid クラスタに接続している。

#### 3.1.1. バックアップ場所のオフラインおよびオンライン化

バックアップ場所を手動でオフラインにし、オンラインに戻します。

#### 手順

1. Data Grid への CLI 接続を作成します。
2. **site status** コマンドを使用して、バックアップの場所がオンラインかオフラインかを確認します。

```
[//containers/default]> site status --cache=cacheName --site=NYC
```



#### 注記

**--site** はオプションの引数です。設定されていない場合、CLI はすべてのバックアップ場所を返します。

3. 次のようにバックアップ場所を管理します。
  - **bring-online** コマンドを使用して、バックアップの場所をオンラインにします。

```
[//containers/default]> site bring-online --cache=customers --site=NYC
```

- **take-offline** コマンドを使用して、バックアップの場所をオフラインにします。

```
[//containers/default]> site take-offline --cache=customers --site=NYC
```

詳細と例については、**help site** コマンドを実行してください。

#### 3.1.2. サイト間の状態転送モードの設定

バックアップの場所がオンラインになったことを DataGrid が検出したときに自動的に発生するように、サイト間の状態転送操作を設定できます。または、デフォルトモードを使用することもできます。これは、CLI を介して、または JMX または REST を介して手動で状態転送を実行するものです。

#### 手順

1. Data Grid への CLI 接続を作成します。
2. 次の例のように、**site** コマンドを使用して状態転送モードを設定します。
  - 現在の状態転送モードを取得します。

```
[//containers/default]> site state-transfer-mode get --cache=cacheName --site=NYC
"MANUAL"
```

- キャッシュとバックアップの場所の自動状態転送操作を設定します。

```
[//containers/default]> site state-transfer-mode set --cache=cacheName --site=NYC --mode=AUTO
```

## ヒント

詳細と例については、**help site** コマンドを実行してください。

### 3.1.3. バックアップ場所への状態のプッシュ

キャッシュの状態をリモートのバックアップ場所に転送します。

#### 手順

1. Data Grid への CLI 接続を作成します。
2. 次の例のように、**site** コマンドを使用して状態の転送をプッシュします。

```
[//containers/default]> site push-site-state --cache=cacheName --site=NYC
```

詳細と例については、**help site** コマンドを実行してください。

#### 参照資料

[Data Grid コマンドラインインターフェイス](#)

## 3.2. REST API を使用したクロスサイト操作の実行

Data Grid サーバーは、クロスサイト操作を実行できるようにする REST API を提供します。

### 3.2.1. すべてのバックアップロケーションのステータス取得

**GET** リクエストですべてのバックアップロケーションのステータスを取得します。

```
GET /v2/caches/{cacheName}/x-site/backups/
```

Data Grid は、以下の例のように、各バックアップロケーションのステータスを JSON 形式で応答します。

```
{
  "NYC": "online",
  "LON": "offline"
}
```

表3.1 リターンステータス

値	説明
online	ローカルクラスター内のすべてのノードには、バックアップの場所を含むクロスサイトビューがあります。
offline	ローカルクラスター内のノードには、バックアップの場所とのクロスサイトビューがありません。
mixed	ローカルクラスター内の一部のノードにはバックアップの場所を含むクロスサイトビューがあり、ローカルクラスター内の他のノードにはクロスサイトビューがありません。応答は、各ノードのステータスを示します。

### 3.2.2. 特定のバックアップ場所のステータスの取得

**GET** リクエストでバックアップロケーションのステータスを取得する。

```
GET /v2/caches/{cacheName}/x-site/backups/{siteName}
```

Data Grid は、以下の例のように、サイト内の各ノードのステータスを JSON 形式で応答します。

```
{
  "NodeA":"offline",
  "NodeB":"online"
}
```

表3.2 リターンステータス

値	説明
online	ノードはオンラインです。
offline	ノードはオフラインです。
failed	ステータスを取得できません。リモートキャッシュがシャットダウンしているか、リクエスト中にネットワークエラーが発生した可能性があります。

### 3.2.3. バックアップ先をオフラインにする

**POST** リクエストと **?action=take-offline** パラメーターを使用して、バックアップの場所をオフラインにします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=take-offline
```

### 3.2.4. バックアップ場所をオンラインにする

**?action=bring-online** パラメーターを使用してバックアップ場所をオンラインにします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=bring-online
```

### 3.2.5. バックアップ場所への状態のプッシュ

**?action=start-push-state** パラメーターを使用して、キャッシュ状態をバックアップ場所にプッシュします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=start-push-state
```

### 3.2.6. 状態転送のキャンセル

**?action=cancel-push-state** パラメーターを使用して状態転送操作をキャンセルします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=cancel-push-state
```

### 3.2.7. 状態転送ステータスの取得

**?action=push-state-status** パラメーターを使用して状態転送操作のステータスを取得します。

```
GET /v2/caches/{cacheName}/x-site/backups?action=push-state-status
```

Data Grid は、以下の例のように、各バックアップ拠点の状態移行の状況を JSON 形式で応答します。

```
{
  "NYC": "CANCELED",
  "LON": "OK"
}
```

表3.3 リターンステータス

値	説明
<b>SENDING</b>	バックアップ場所への状態転送が進行中です。
<b>OK</b>	状態の転送が正常に完了しました。
<b>ERROR</b>	状態転送でエラーが発生しました。ログファイルを確認してください。
<b>CANCELLING</b>	状態移行のキャンセルが進行中です。

### 3.2.8. 状態転送ステータスのクリア

**?action=clear-push-state-status** パラメーターを使用して送信サイトの状態転送ステータスをクリアします。

-

```
POST /v2/caches/{cacheName}/x-site/local?action=clear-push-state-status
```

### 3.2.9. オフラインテイク条件の変更

特定の条件が満たされると、サイトはオフラインになります。オフラインにするパラメーターを変更して、バックアップロケーションが自動的にオフラインになるタイミングを制御します。

#### 手順

1. **GET** リクエストと **take-offline-config** パラメーターで設定されたテイクオフラインパラメーターを確認します。

```
GET /v2/caches/{cacheName}/x-site/backups/{siteName}/take-offline-config
```

Data Grid のレスポンスには、以下のように **after\_failures** と **min\_wait** フィールドがあります。

```
{
  "after_failures": 2,
  "min_wait": 1000
}
```

2. **PUT** リクエストの本文のオフライン取得パラメーターを変更します。

```
PUT /v2/caches/{cacheName}/x-site/backups/{siteName}/take-offline-config
```

操作が正常に完了すると、サービスは **204 (No Content)** を返します。

### 3.2.10. 受信サイトからの状態転送のキャンセル

2つのバックアップ場所間の接続が切断された場合、プッシュを受信しているサイトでの状態転送をキャンセルできます。

**?action=cancel-receive-state** パラメーターで、リモートサイトからの状態転送をキャンセルし、ローカルキャッシュの現在の状態を維持する。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=cancel-receive-state
```

### 3.2.11. バックアップ場所のステータスの取得

**GET** 要求により、キャッシュ・マネージャーからすべてのバックアップ・ロケーションのステータスを取得します。

```
GET /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/
```

Data Grid は、以下の例のように JSON 形式でステータスを応答します。

```
{
  "SFO-3":{
    "status":"online"
  },
  "NYC-2":{
```

```

    "status":"mixed",
    "online":[
      "CACHE_1"
    ],
    "offline":[
      "CACHE_2"
    ]
  }
}

```

表3.4 リターンステータス

値	説明
<b>online</b>	ローカルクラスター内のすべてのノードには、バックアップの場所を含むクロスサイトビューがあります。
<b>offline</b>	ローカルクラスター内のノードには、バックアップの場所とのクロスサイトビューがありません。
<b>mixed</b>	ローカルクラスター内の一部のノードにはバックアップの場所を含むクロスサイトビューがあり、ローカルクラスター内の他のノードにはクロスサイトビューがありません。応答は、各ノードのステータスを示します。

### 3.2.12. バックアップ先をオフラインにする

**?action=take-offline** パラメーターで、バックアップロケーションをオフラインにします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=take-offline
```

### 3.2.13. バックアップ場所をオンラインにする

**?action=bring-online** パラメーターを使用してバックアップ場所をオンラインにします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=bring-online
```

### 3.2.14. 状態転送モードの取得

**GET** リクエストで状態転送モードを確認してください。

```
GET /rest/v2/caches/{cacheName}/x-site/backups/{site}/state-transfer-mode
```

### 3.2.15. 状態転送モードの設定

**?action=set** パラメーターを使用して状態転送モードを設定します。



```
POST /rest/v2/caches/{cacheName}/x-site/backups/{site}/state-transfer-mode?action=set&mode={mode}
```

### 3.2.16. 状態転送の開始

**?action=start-push-state** パラメーターを使用して、すべてのキャッシュの状態をリモートサイトにプッシュします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=start-push-state
```

### 3.2.17. 状態転送のキャンセル

**?action=cancel-push-state** パラメーターを使用して、進行中の状態転送操作をキャンセルします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=cancel-push-state
```

## 3.3. JMX を使用したクロスサイト操作の実行

Data Grid は、状態転送のプッシュやサイトをオンラインにするなど、クロスサイト操作を実行する JMX ツールを提供します。

### 3.3.1. JMX MBean を登録するための Data Grid の設定

Data Grid は、統計の収集と管理操作の実行に使用できる JMX MBean を登録できます。JMX とは別に統計を有効にする必要があります。そうしないと、Data Grid はすべての統計属性に **0** の値を提供しません。

#### 手順

以下のいずれかの方法でキャッシュコンテナ設定を変更し、JMX を有効にします。

- 宣言型: `<jmx enabled="true" />` 要素をキャッシュコンテナに追加します。
- プログラマティック: `.jmx().enable()` メソッドを呼び出します。

#### 宣言型

```
<cache-container>
  <jmx enabled="true" />
</cache-container>
```

#### プログラマティック

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
    .jmx().enable()
    .build();
```

### 3.3.2. クロスサイト操作の実行

JMX クライアント経由でクロスサイト操作を実行します。

### 前提条件

- JMX MBean を登録するように Data Grid を設定します。

### 手順

1. 任意の JMX クライアントで Data Grid に接続します。
2. 以下の MBean から操作を呼び出します。
  - **XSiteAdmin** は、キャッシュのクロスサイト操作を提供します。
  - **GlobalXSiteAdminOperations** は、Cache Manager のクロスサイト操作を提供します。  
たとえば、サイトをオンラインに戻すには、**bringSiteOnline(siteName)** を呼び出します。

利用可能なクロスサイト操作の詳細については、**Data Grid JMX Components**のドキュメントを参照してください。

### 参照資料

- [XSiteAdmin MBean](#)
- [GlobalXSiteAdminOperations MBean](#)

## 第4章 グローバル DATA GRID クラスターの監視およびトラブルシューティング

Data Grid は、JMX または Data Grid Server の **/metrics** エンドポイント経由で、クロスサイトレプリケーション操作の統計を提供します。

クロスサイトレプリケーションの統計はキャッシュレベルで利用できるため、キャッシュの統計を明示的に有効にする必要があります。同様に、JMX を介して統計を収集する場合は、Data Grid を設定して MBean を登録する必要があります。

Data Grid には **org.infinispan.XSITE** ログカテゴリも含まれるため、ネットワークおよび状態遷移操作に関する一般的な問題を監視し、トラブルシューティングすることができます。

### 4.1. DATA GRID 統計の有効化

Cache Manager およびキャッシュの統計をエクスポートするように Data Grid を設定します。

#### 手順

以下のいずれかの方法で、Data Grid 統計が有効化されるように設定を変更します。

- 宣言型: **statistics="true"** 属性を追加します。
- プログラマティック: **.statistics()** メソッドを呼び出します。

#### 宣言型

```
<!-- Enables statistics for the Cache Manager. -->  
<cache-container statistics="true">  
  <!-- Enables statistics for the named cache. -->  
  <local-cache name="mycache" statistics="true"/>  
</cache-container>
```

#### プログラマティック

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()  
  //Enables statistics for the Cache Manager.  
  .cacheContainer().statistics(true)  
  .build();  
  
Configuration config = new ConfigurationBuilder()  
  //Enables statistics for the named cache.  
  .statistics().enable()  
  .build();
```

### 4.2. DATA GRID メトリクスの設定

Data Grid を設定して、**metrics** エンドポイント経由でゲージとヒストグラムをエクスポートします。

#### 手順

- 必要に応じて、**metrics** 設定でゲージとヒストグラムをオンまたはオフにします。

## 宣言型

```
<!-- Computes and collects statistics for the Cache Manager. -->
<cache-container statistics="true">
  <!-- Exports collected statistics as gauge and histogram metrics. -->
  <metrics gauges="true" histograms="true" />
</cache-container>
```

## プログラマティック

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
  //Computes and collects statistics for the Cache Manager.
  .statistics().enable()
  //Exports collected statistics as gauge and histogram metrics.
  .metrics().gauges(true).histograms(true)
  .build();
```

### 4.2.1. Data Grid メトリクスの収集

Prometheus などのモニターリングツールを使用して、Data Grid メトリクスを収集します。

#### 前提条件

- 統計を有効にします。統計を有効にしないと、Data Grid はメトリクスに **0** と **-1** の値を指定します。
- 必要に応じて、ヒストグラムを有効にします。デフォルトでは、Data Grid はゲージを生成しますが、ヒストグラムは生成しません。

#### 手順

- Prometheus (OpenMetrics) 形式でメトリクスを取得します。

```
$ curl -v http://localhost:11222/metrics
```

- MicroProfile JSON 形式でメトリクスを取得します。

```
$ curl --header "Accept: application/json" http://localhost:11222/metrics
```

#### 次のステップ

Data Grid メトリクスを収集するようにモニターリングアプリケーションを設定します。たとえば、以下を **prometheus.yml** に追加します。

```
static_configs:
  - targets: ['localhost:11222']
```

#### 参照資料

- [Prometheus Configuration](#)
- Data Grid 統計の有効化

## 4.3. JMX MBEAN を登録するための DATA GRID の設定

Data Grid は、統計の収集と管理操作の実行に使用できる JMX MBean を登録できます。JMX とは別に統計を有効にする必要があります。そうしないと、Data Grid はすべての統計属性に **0** の値を提供しません。

### 手順

以下のいずれかの方法でキャッシュコンテナ設定を変更し、JMX を有効にします。

- 宣言型: `<jmx enabled="true" />` 要素をキャッシュコンテナに追加します。
- プログラマティック: `.jmx().enable()` メソッドを呼び出します。

### 宣言型

```
<cache-container>
  <jmx enabled="true" />
</cache-container>
```

### プログラマティック

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
    .jmx().enable()
    .build();
```

### 4.3.1. クロスサイトレプリケーション用の JMX MBean

Data Grid は、統計を収集し、リモート操作を実行できるクロスサイトレプリケーションに JMX MBean を提供します。

**org.infinispan:type=Cache** コンポーネントは、以下の JMX MBean を提供します。

- **XSiteAdmin** は、特定のキャッシュインスタンスに適用されるクロスサイト操作を公開しません。
- **StateTransferManager** は、状態遷移操作の統計を提供します。
- **InboundInvocationHandler** は、非同期および同期のクロスサイトリクエストの統計と操作を提供します。

**org.infinispan:type=CacheManager** コンポーネントには以下の JMX MBean が含まれます。

- **GlobalXSiteAdminOperations** は、キャッシュコンテナのすべてのキャッシュに適用されるクロスサイト操作を公開します。

JMX MBean および利用可能な操作および統計の説明に関する詳細は、**Data Grid JMX Components** のドキュメントを参照してください。

### 参照資料

[Data Grid JMX Components](#)

## 4.4. ログの収集およびクロスサイトレプリケーションのトラブルシューティング

Data Grid のクロスサイトレプリケーションに関連する問題を診断し、解決します。Data Grid コマンドラインインターフェイス (CLI) を使用して、実行時にログレベルを調整し、クロスサイトのトラブルシューティングを実行します。

### 手順

1. `$RHDG_HOME` でターミナルを開きます。
2. Data Grid CLI コネクションを作成します。
3. 必要な場合には、ランタイムのロギングレベルを調整して DEBUG メッセージをキャプチャします。  
たとえば、以下のコマンドは、`org.infinispan.XSITE` カテゴリの DEBUG ログメッセージを有効にします。

```
[//containers/default]> logging set --level=DEBUG org.infinispan.XSITE
```

次に、`#{rhdg.server.root}/log` ディレクトリーで、クロスサイトメッセージの Data Grid ログファイルを確認できます。

4. `site` コマンドを使用して、バックアップの場所のステータスを表示し、トラブルシューティングを実行します。

たとえば、バックアップの場所に "LON" を使用する "customers" キャッシュのステータスを確認します。

```
[//containers/default]> site status --cache=customers
{
  "LON" : "online"
}
```

Data Grid CLI を使用してトラブルシューティングするもう1つのシナリオは、状態遷移操作中にバックアップの場所間のネットワーク接続が切断された場合です。

この場合、状態遷移を受信する Data Grid クラスタは、操作が完了するまで継続的に待機します。この場合、受信サイトへの状態遷移をキャンセルして、通常の運用状態に戻す必要があります。

たとえば、以下のように NYC の状態遷移をキャンセルします。

```
[//containers/default]> site cancel-receive-state --cache=mycache --site=NYC`
```

### 参照資料

- [Data Grid Server Troubleshooting](#)
- [Working with Data Grid Server Logs](#)

#### 4.4.1. クロスサイトログメッセージ

クロスサイトレプリケーションに関連するログメッセージのユーザーアクションを検索します。

ログレベル	識別子	メッセージ	説明
DEBUG	ISPN000400	ノード null が疑われました	Data Grid は、バックアップの場所に到達できない場合にこのメッセージを出力します。サイトがオンラインであることを確認し、ネットワークステータスを確認します。
INFO	ISPN000439	新しい x-site ビューの受信: \${site.name}	Data Grid は、サイトがグローバルクラスターに参加および離脱するときに、このメッセージを出力します。
INFO	ISPN100005	サイト \${site.name} はオンラインです。	Data Grid は、サイトがオンラインになると、このメッセージを出力します。
INFO	ISPN100006	サイト \${site.name} はオフラインです。	Data Grid は、サイトがオフラインになると、このメッセージを出力します。サイトを手動でオフラインにしなかった場合、このメッセージは障害が発生したことを示している可能性があります。ネットワークのステータスを確認し、サイトをオンラインに戻してみてください。
WARN	ISPN000202	キャッシュ \${cache.name} のデータをサイト \${site.name} にバックアップする際の問題:	Data Grid は、例外とともに状態転送操作で問題が発生した場合に、このメッセージを出力します。必要な場合は、Data Grid のロギングを調整して、より詳細なロギングメッセージを取得します。
WARN	ISPN000289	X-Site 状態チャンクを \${site.name} に送信できません。	状態遷移操作中に Data Grid がキャッシュエントリーのバッチを転送できないことを示します。サイトがオンラインであることを確認し、ネットワークステータスを確認します。
WARN	ISPN000291	X-Site 状態チャンクを適用できません。	状態遷移操作中に Data Grid がキャッシュエントリーのバッチを適用できないことを示します。サイトがオンラインであることを確認し、ネットワークステータスを確認します。

ログレベル	識別子	メッセージ	説明
WARN	ISPN000322	サイト <code>{site.name}</code> への <code>x</code> サイトの状態遷移を再開できません。	Data Grid が、バックアップ場所への状態遷移操作を再開できないことを示します。サイトがオンラインであることを確認し、ネットワークステータスを確認します。
ERROR	ISPN000477	サイト <code>{site.name}</code> に対して操作 <code>{operation.name}</code> を実行できません	Data Grid がバックアップの場所で操作を正常に完了できないことを示します。必要な場合は、Data Grid のロギングを調整して、より詳細なロギングメッセージを取得します。
FATAL	ISPN000449	Xsite の状態遷移タイムアウトは、1 以上でなければなりません。	<b>timeout</b> 属性の値が <b>0</b> または負の値である場合の結果です。キャッシュ定義の状態遷移設定の <b>timeout</b> 属性に <b>1</b> 以上の値を指定します。
FATAL	ISPN000450	再試行間の Xsite 状態遷移の待機時間は、1 以上である必要があります。	<b>wait-time</b> 属性の値が <b>0</b> または負の値である場合の結果です。キャッシュ定義の状態遷移設定の <b>wait-time</b> 属性に <b>1</b> 以上の値を指定します。
FATAL	ISPN000576	ローカルキャッシュでクロスサイトレプリケーションは利用できません。	クロスサイトレプリケーションは、ローカルキャッシュモードでは機能しません。ローカルキャッシュ定義からバックアップ設定を削除するか、分散またはレプリケートされたキャッシュモードを使用します。